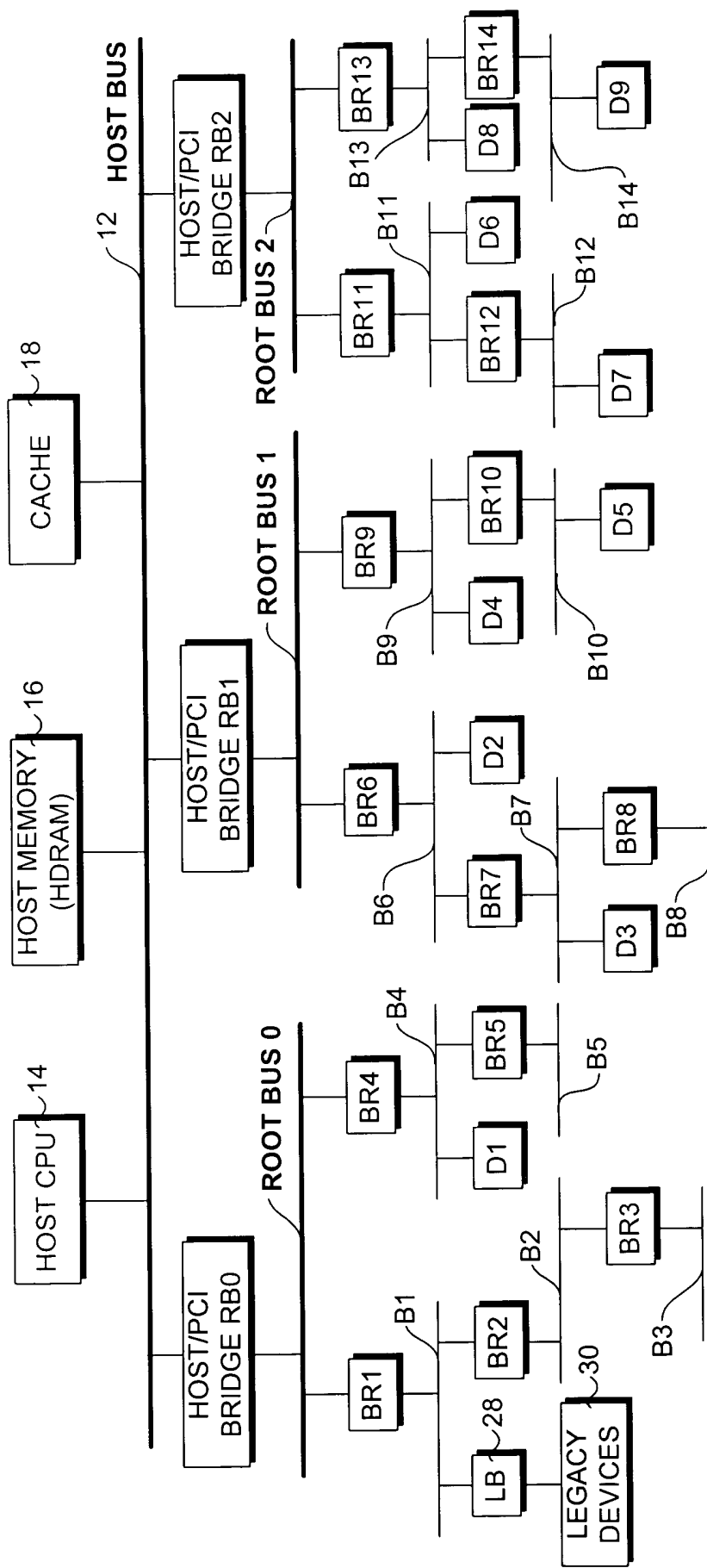
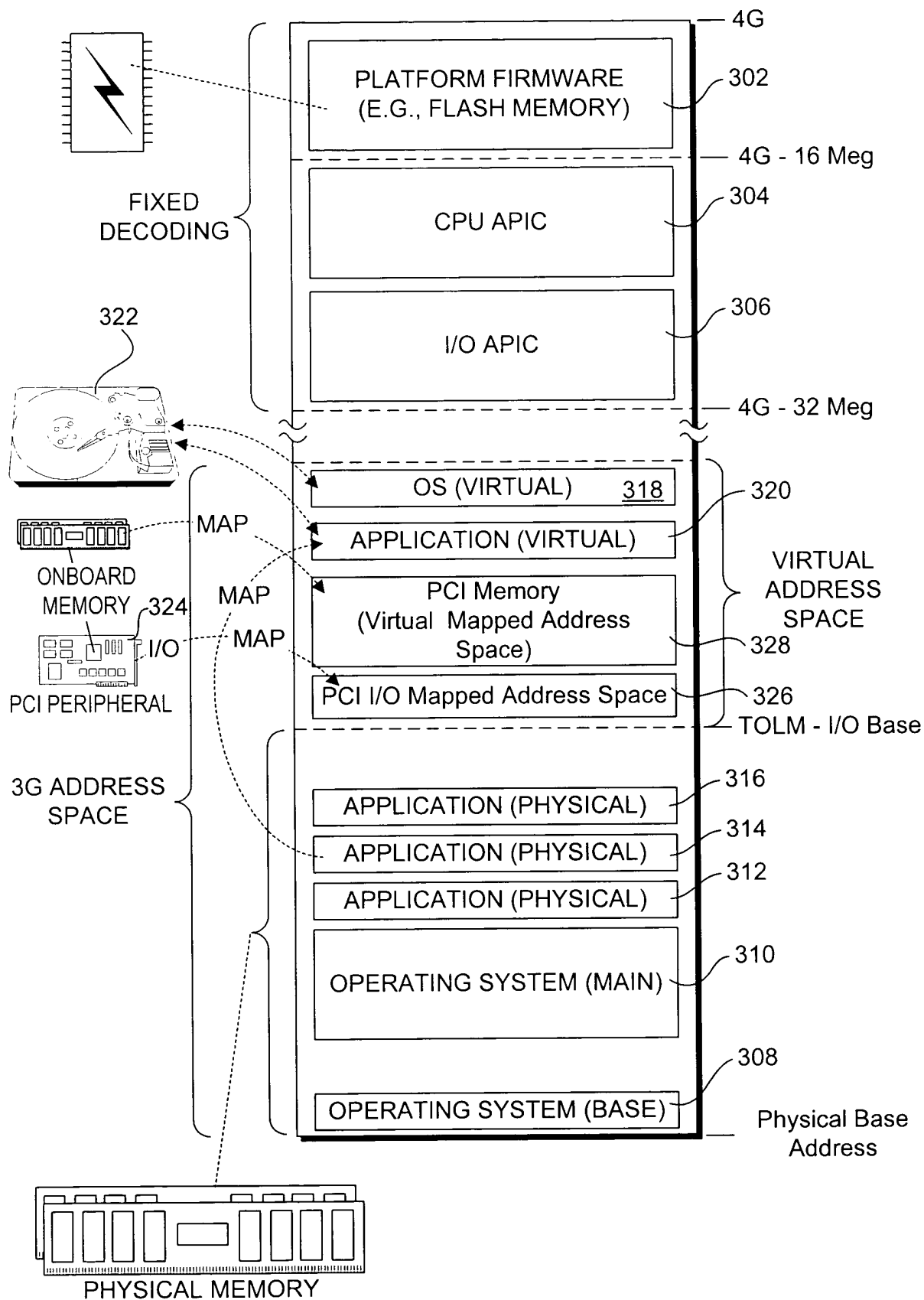


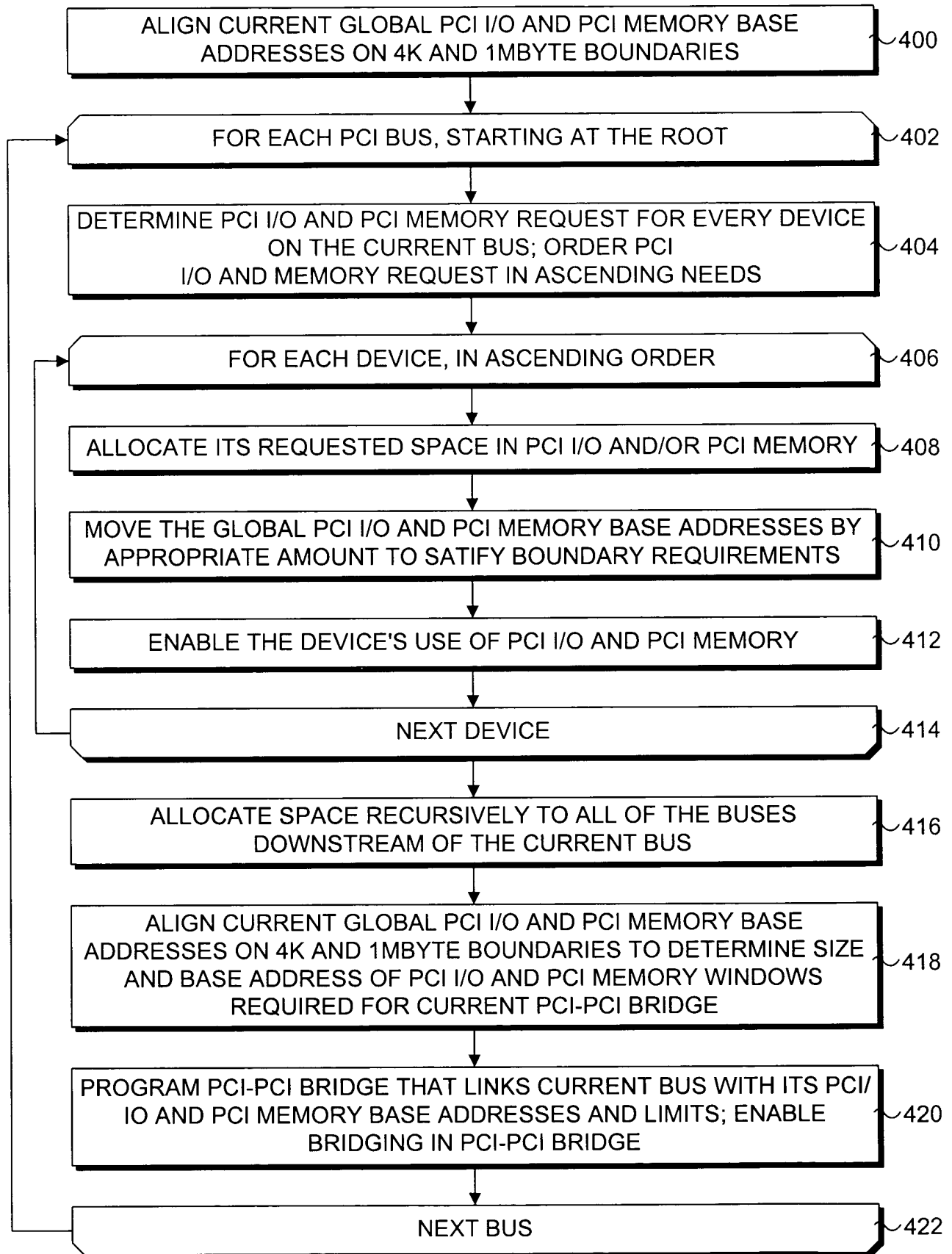
**Fig. 1**



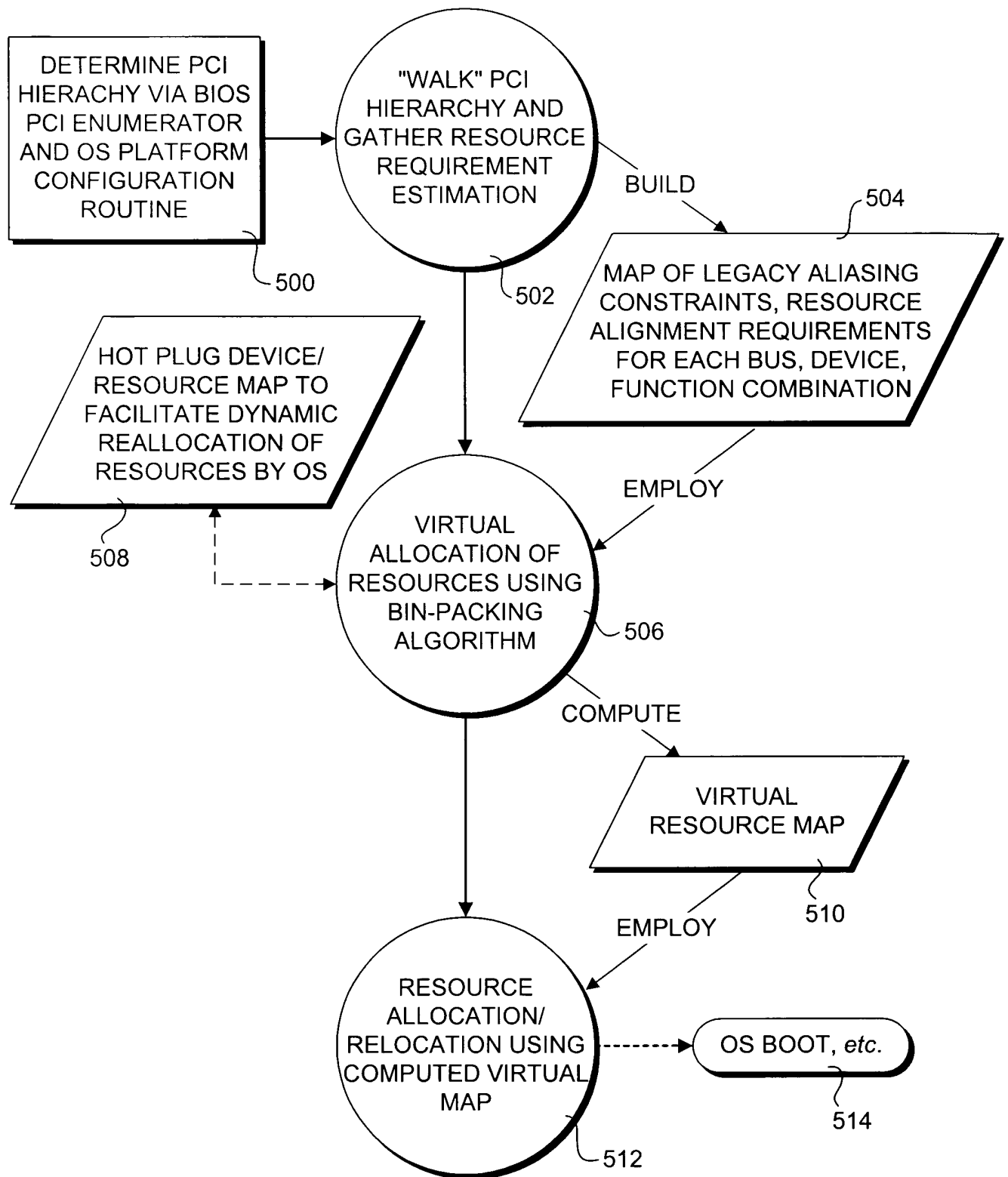
*Fig. 2*



**Fig. 3**

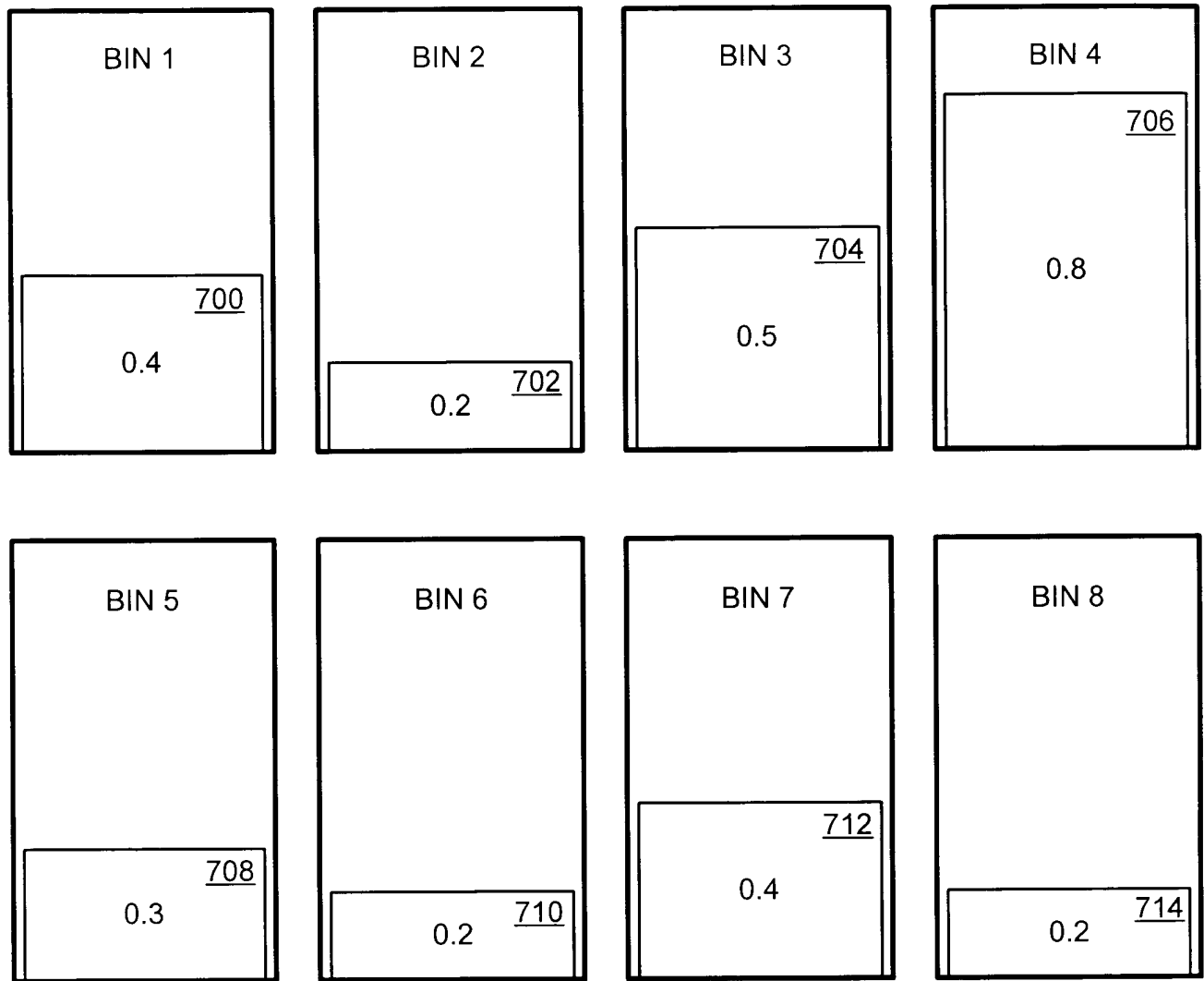


***Fig. 4 (Prior Art)***

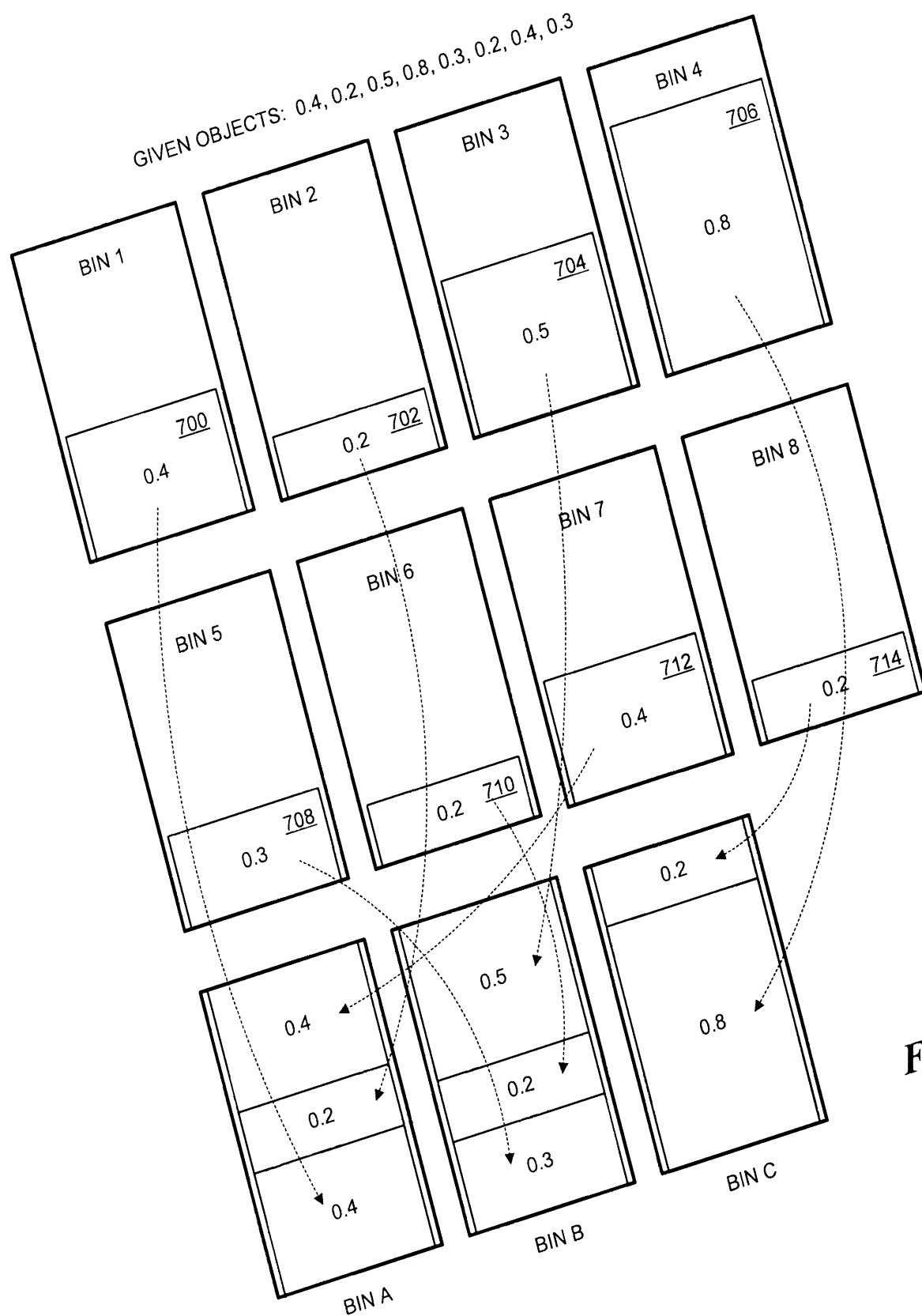


**Fig. 5**

GIVEN OBJECTS: 0.4, 0.2, 0.5, 0.8, 0.3, 0.2, 0.4, 0.2



***Fig. 6 (Prior Art)***



**Fig. 7**

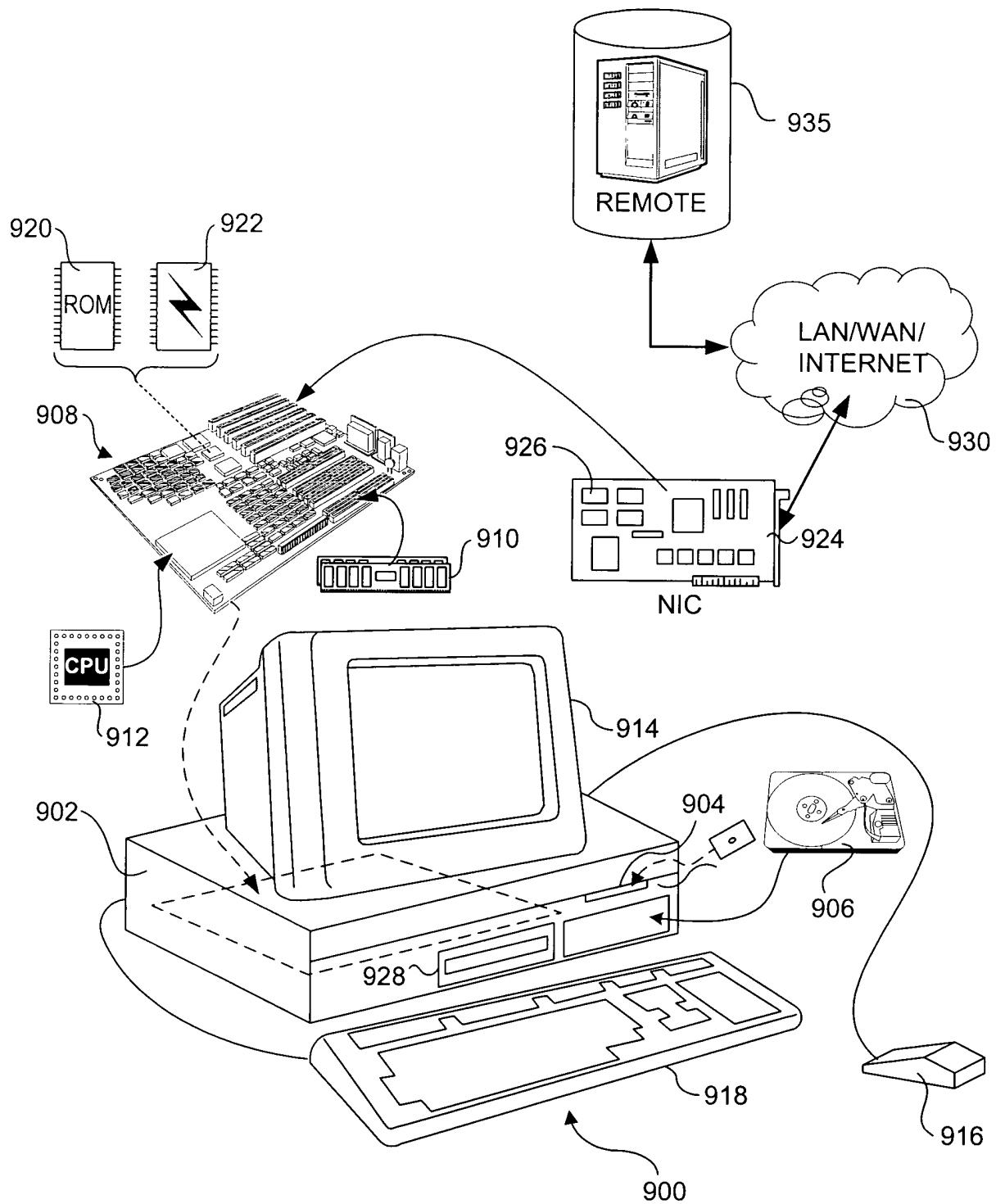
```

knapSack_k(C,S) // S an array with n values (the sizes)
  int maxSum, sum, j; // C is the size of the knapsack
  Set T = new Set;
  take =  $\emptyset$ ; maxSum = 0;
  for each subset  $T \subseteq \{1, \dots, n\}$ 
    with k elements {
      sum = the sum of the values S[i] such that  $i \in T$ ;
      if (sum  $\leq$  C) {
        for each j not in T {
          if (sum + S[j]  $\leq$  C){
            sum = sum + S[j];
            T = T  $\cup$  {j};}
        }
        if (maxSum < sum) {maxSum = sum}
      }
    }
  } return maxSum;

```

*Fig. 8*





**Fig. 9**